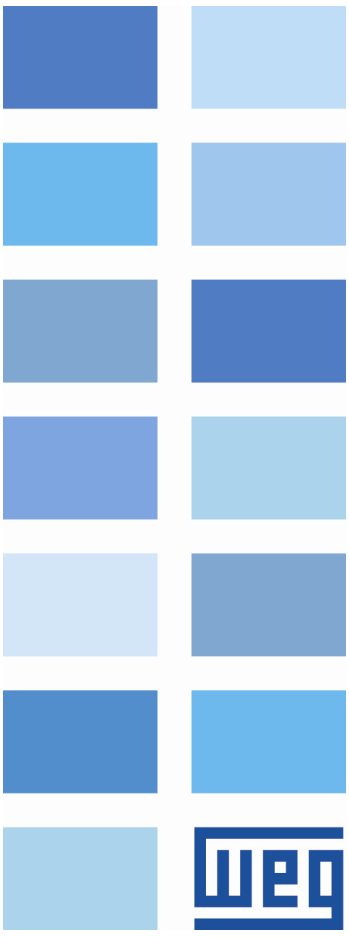
Serial

CFW-11

Manual de la Comunicación

Idioma: Español



Manual de la Comunicación Serial RS232 / RS485

Serie: CFW-11

Idioma: Español

N° del Documento: 0899.5742 / 03

Fecha de la Publicación: 01/2010

Índice

A RESPECTO DEL MANUAL 5

Abreviaciones y Definiciones 5
Representación Numérica 5

1 INTRODUCCIÓN A LA COMUNICACIÓN SERIAL 6

2 KITS ACCESORIOS 7

2.1 RS232 7
2.1.1 Kit RS232-01 7
2.1.2 Terminales del Conector 7
2.1.3 Señalizaciones y Llaves 7
2.1.4 Conexión con la Red RS232 7
2.1.5 Cables para Conexión en RS232 8
2.2 RS485 8
2.2.1 Kit RS485-01 8
2.2.2 Kit CAN/RS485-01 8
2.2.3 Terminales del Conector 9
2.2.4 Señalizaciones y Llaves 9
2.2.5 Conexiones con la red RS485 9
2.3 ANYBUS-CC 9

3 PARAMETRIZACIÓN DEL CONVERTIDOR 10

3.1 Símbolos para Descripción de las Propiedades 10
P0105 – Selección 1ª/2ª Rampa 10
P0308 – Dirección Serial 10
P0310 – Tasa de Comunicación Serial 11
P0311 – Configuración de los Bytes de la Interface Serial 11
P0312 – Protocolo Serial 11
P0313 – Acción para Error de Comunicación 12
P0314 – Watchdog Serial 12
P0316 – Estado de la Interface Serial 13
P0680 – Estado Lógico 13
P0681 – Velocidad del Motor en 13 bits 14
P0682 – Palabra de Control vía Serial / USB 15
P0683 – Consigna de Velocidad vía Serial/ USB 16
P0695 – Valor para las Salidas Digitales 16
P0696 – Valor 1 para Salidas Analógicas 17
P0697 – Valor 2 para Salidas Analógicas 17
P0698 – Valor 3 para Salidas Analógicas 17
P0699 – Valor 4 para Salidas Analógicas 17

4 PROTOCOLO TP 19

4.1 Campos do Protocolo 19
4.2 Formato de los Telegramas 20
4.2.1 Telegrama de Lectura 20
4.2.2 Telegrama de Escrita 20
4.3 Ejemplo de Telegramas Utilizando el Protocolo TP 20

5 PROTOCOLO MODBUS-RTU 22

5.1 Modos de Transmisión 22
5.2 Estructura de los Mensajes en el Modo RTU 22
5.2.1 Dirección 22
5.2.2 Código de la Función 22
5.2.3 Campo de Datos 23
5.2.4 CRC 23
5.2.5 Tiempo entre Mensajes 23
5.3 Operación del CFW-11 en la Red MODBUS-RTU 24

5.3.1 Funciones Disponibles y Tiempos de Respuesta 24
5.3.2 Dirección de los Datos y Offset 25
5.4 DESCRIPCIÓN DETALLADA DE LAS FUNCIONES 25
5.4.1 Función 03 – Read Holding Register 25
5.4.2 Función 06 – Write Single Register 26
5.4.3 Función 16 – Write Multiple Registers 26
5.4.4 Función 43 – Read Device Identification 27
5.4.5 Errores de Comunicación 28
6 FALLAS Y ALARMAS RELACIONADAS CON LA COMUNICACIÓN SERIAL 30
I. APÉNDICES 31
APÉNDICE A. TABLA ASCII 31
APÉNDICE B. CÁLCULO DEL CRC UTILIZANDO TABLAS 32
APÉNDICE C. CÁLCULO DEL CRC UTILIZANDO DESPLAZAMIENTO DE REGISTRADORES 33

A Respecto del Manual

Este manual provee la descripción necesaria para la operación del convertidor de frecuencia CFW-11 utilizando las interfaces serie RS232 o RS485. Este manual debe ser utilizado en conjunto con el manual del usuario del CFW-11.

Abreviaciones y Definiciones

ASCII American Standard Code for Information Interchange
CRC Cycling Redundancy Check
EIA Electronic Industries Alliance
RTU Remote Terminal Unit

Representación Numérica

Números decimales son representados a través de dígitos sin sufijo. Números hexadecimales son representados con la letra 'h' después del número.

1 Introducción a la Comunicación Serial

En una interface serial los bits de datos son enviados de modo secuencial a través de un canal de comunicación o bus. Diversas tecnologías utilizan comunicación serial para la transferencia de datos, incluyendo las interfaces RS232 y RS485.

Las normas que especifican los padrones RS232 y RS485, sin embargo, no especifican el formato ni la secuencia de caracteres para la transmisión y recepción de datos. En este sentido, además de la interface, es necesario identificar también el protocolo utilizado para la comunicación. Entre los diversos protocolos existentes, un protocolo muy utilizado en la industria es el protocolo Modbus-RTU.

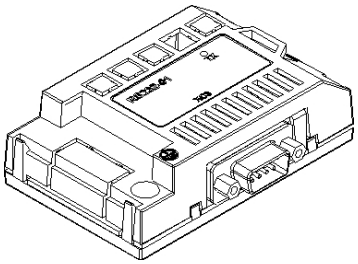
A seguir serán presentadas las características de las interfaces serial RS232 y RS485 disponible para el convertidor CFW-11, así como los protocolos para la utilización de estas interfaces.

2 Kits Accesorios

Para agregar una interface serial para el convertidor de frecuencia CFW-11 es necesario utilizar un de los kits para comunicación RS232 o RS485 descriptos a seguir. Informaciones de cómo proceder la instalación de estos módulos en el convertidor de frecuencia pueden ser obtenidas en el guía rápido que acompaña el kit.

2.1 RS232

2.1.1 Kit RS232-01



- ☑ Ítem WEG: 10051958.
- ☑ Compuesto por el módulo de comunicación RS232 (figura al lado), guía rápido de montaje y tornillo de fijación.
- ☑ Interface sigue el padrón EIA RS232C.
- ☑ Permite la conexión entre el CFW-11 y el maestro de la red (punto a punto).
- ☑ Longitud máxima para conexiones de los dispositivos: 10 metros.

2.1.2 Terminales del Conector

El módulo para comunicación RS232 posee un conector DB9 macho (XC8) con los siguientes pines:

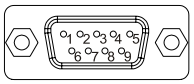


Tabla 2.1 *- Pines del conector DB9 para RS232*

Pines	Nombre	Función
1	No conectado	-
2	RX	Recepción de datos
3	TX	Transmisión de datos
4	No conectado	-
5	GND	Referencia para circuito RS232
6	No conectado	-
7	No conectado	-
8	No conectado	-
9	No conectado	-

2.1.3 Señalizaciones y Llaves

☑ **LED TX:** LED para señalizar la transmisión de datos por el convertidor de frecuencia, en el color verde.

2.1.4 Conexión con la Red RS232

- ☑ Los señales RX y TX del convertidor deben ser conectadas respectivamente a los señales TX y RX del maestro, además de la conexión de lo señal de referencia (GND).
- ☑ La interface RS232 es muy susceptible a interferencias. Por este motivo, el cable utilizado para la comunicación debe ser el más corto posible – siempre menor que 10 metros – y debe ser puesto por separado del cableado de potencia que alimenta el convertidor de frecuencia y el motor.

2.1.5 Cables para Conexión en RS232

Caso sea deseado, la WEG puede suministrar los siguientes cables para conexión en RS232 entre el convertidor CFW-11 y un maestro de la red, como una Computadora:

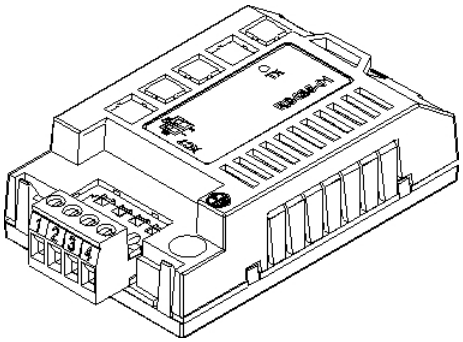
Cable	Ítem WEG
Cable RS232 apantallado con conectores DB9 hembra Longitud: 3 metros	10050328
Cable RS232 apantallado con conectores DB9 hembra Longitud: 10 metros	10191117

Otros cables, sin embargo, pueden ser encontrados en el mercado – en general denominados *null-modem* – o fabricados de acuerdo con el deseado para la instalación.

2.2 RS485

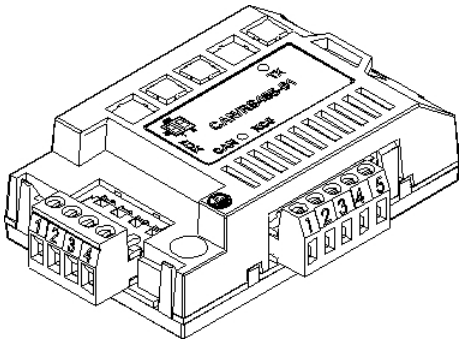
El CFW-11 posee dos opciones para utilizar la interface RS485, descritas a seguir.

2.2.1 Kit RS485-01



- ☑ Ítem WEG: 10051957.
- ☑ Compuesto por el módulo de comunicación RS485 (figura a lado), guía rápido de montaje y tornillo de fijación.
- ☑ Interface sigue el padrón EIA-485.
- ☑ Interface aislada galvánicamente y con señal diferencial, confiriendo mayor robustez contra interferencia electromagnética.
- ☑ Permite la conexión de hasta 32 dispositivos en el mismo segmento. Una cuantidad mayor de dispositivos puede ser conectada con el uso de repetidores.[1]
- ☑ Longitud máximo del bus: 1000 metros.

2.2.2 Kit CAN/RS485-01



- ☑ Ítem WEG: 10051960.
- ☑ Compuesto por el módulo de comunicación CAN/RS485-01 (figura a lado), guía rápido de montaje y tornillo de fijación.
- ☑ Posee las mismas características de la interface RS485-01, más una interface CAN, para aplicaciones donde sea necesario la operación en conjunto de ambas las interfaces.

[1] El número límite de equipos que pueden ser conectados en la red también depende del protocolo utilizado.

2.2.3 Terminales del Conector

El módulo para comunicación RS485 posee un conector *plug-in* de 4 vías (XC7) con los siguiente pines:

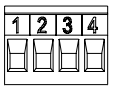
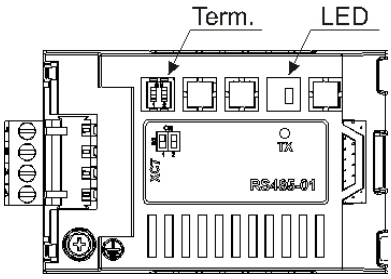


Tabla 2.2 *- Pines del conector de 4 vías para RS485*

Pines	Nombre	Función
1	A-Line (-)	RxD/TxD negativo
2	B-Line (+)	RxD/TxD positivo
3	GND	0V aislado del circuito RS485
4	Ground	Tierra (blindaje)

2.2.4 Señalizaciones y Llaves



- ☑ **LED TX:** LED para señalizar la transmisión de datos por el convertidor, en el color verde.
- ☑ **Resistor de terminación (S1):** llave para habilitar el resistor de terminación necesario para la interface RS485. Este resistor debe ser habilitado (posición *ON*) solamente en los dos dispositivos ubicados en los extremos del bus principal.

2.2.5 Conexiones con la red RS485

Para la conexión del convertidor utilizando la interface RS485, los siguientes puntos deben ser observados:

- ☑ Es recomendado el uso de un cable con par tranzado blindado.
- ☑ Se recomienda también que el cable posee más un conductor para la conexión de la señal de referencia (GND). Caso el cable no posea el conductor adicional, se debe dejar la señal GND desconectado.
- ☑ La instalación del cable debe ser separado (y si posible lejos) del cableados de potencia.
- ☑ Todos los dispositivos de la red deben estar debidamente puestos a tierra, de preferencia en la misma conexión con a tierra. El blindaje del cable también debe ser puesto a tierra.
- ☑ Habilitar los resistores de terminación solo en dos puntos, en los extremos del bus principal, mismo que existan derivaciones a partir del bus.

2.3 ANYBUS-CC

Las interfaces RS232 y RS485 también pueden ser agregadas utilizando los kits Anybus-CC pasivos para RS232 o RS485. Consulte el Manual de la Comunicación Anybus-CC para obtener informaciones a respecto de estos kits.

3 Parametrización del Convertidor

A seguir es presentado solo os parámetros del convertidor de frecuencia CFW-11 que poseen relación con la comunicación serial.

3.1 Símbolos para Descripción de las Propiedades

RO	Parámetro solamente de lectura.
CFG	Parámetro solamente modificado con el motor parado.
Net	Parámetro visible a través de la HMI si el convertidor posee interface de red instalada – RS232, RS485, CAN, Anybus-CC, Profibus – o si la interface USB fuera conectada.
Serial	Parámetro visible a través de la HMI si el convertidor poseer interface RS232 o RS485 instalada.
USB	Parámetro visible a través de la HMI si la interface USB del convertidor fuera conectada.

P0105 – Selección 1ª/2ª Rampa

P0220 – Selección LOCAL/REMOTA

P0221 – Selección de la Referencia de Velocidad – Situación LOCAL

P0222 – Selección de la Referencia de Velocidad – Situación REMOTA

P0223 – Selección del Sentido de Giro – Situación LOCAL

P0224 – Selección de Gira / Para – Situación LOCAL

P0225 – Selección de JOG – Situación LOCAL

P0226 – Selección del Sentido de Giro – Situación REMOTA

P0227 – Selección de Gira / Para – Situación REMOTA

P0228 – Selección de JOG – Situación REMOTA

Estos parámetros son utilizados en la configuracións de la fuente de los comandos para los modos local y remoto del convertidor de frecuencia CFW-11. Para que el convertidor de frecuencia sea controlado a través de la interface Serial, se debe selecionar una de las opciones 'Serial/USB' disponíbles en los parámetros.

La descripción detallada de estos parámetros se encuentra en el Manual de programación del CFW-11.

P0308 – Dirección Serial

Rango: 1 … 247 **Padrón:** 1

Propiedades: CFG, Serial

Grupos de acceso vía HMI:

01 GRUPOS PARÁMETROS
- 49 Comunicación
  - 113 Serial RS232/485

Descripción:
Permite programar la dirección utilizado para comunicación serial del convertidor. Es necesario que cada equipo de la red posea una dirección distinta de los demás. Las direcciones válidas para este parámetro dependen del protocolo programado en el P0312:

☑ P0312 = 1 (TP) → direcciones válidas: 1 … 30.
☑ P0312 = 2 (Modbus-RTU) → direcciones válidas: 1 … 247.

P0310 – Tasa de Comunicación Serial

Rango: 0 = 9600 bits/s
1 = 19200 bits/s
2 = 38400 bits/s
3 = 57600 bits/s

Padrón: 0

Propiedades: CFG, Serial

Grupos de acceso vía HMI:

01 GRUPOS PARÁMETROS
∟ 49 Comunicación
∟ 113 Serial RS232/485

Descripción:
Permite programar el valor deseado para la tasa de comunicación de la interface serial, en bits por segundo. Esta tasa debe ser la misma para todos los equipos conectados en la red.

P0311 – Configuración de los Bytes de la Interface Serial

Rango: 0 = 8 bits de datos, sin paridad, 1 stop bit
1 = 8 bits de datos, paridad par, 1 stop bit
2 = 8 bits de datos, paridad impar, 1 stop bit
3 = 8 bits de datos, sin paridad, 2 stop bits
4 = 8 bits de datos, paridad par, 2 stop bits
5 = 8 bits de datos, paridad impar, 2 stop bits

Padrón: 0

Propiedades: CFG, Serial

Grupos de acceso vía HMI:

01 GRUPOS PARÁMETROS
∟ 49 Comunicación
∟ 113 Serial RS232/485

Descripción:
Permite la configuración del número de bits de datos, paridad y *stop* bits en los bytes de la interface serial. Esta configuración debe ser la misma para todos los equipos conectados en la red.

P0312 – Protocolo Serial

Rango: 1 = TP
2 = Modbus-RTU

Padrón: 2

Propiedades: CFG, Serial

Grupos de acceso vía HMI:

01 GRUPOS PARÁMETROS
∟ 49 Comunicación
∟ 113 Serial RS232/485

Descripción:
Permite seleccionar el protocolo deseado para la interface serial. La descripción detallada de los protocolos es hecha en los ítems siguientes de este manual.

P0313 – Acción para Error de Comunicación

Rango: 0 = Inactivo
1 = Para por Rampa
2 = Deshabilita General
3 = Va para Modo Local
4 = Va para Modo Local y mantenga comandos y referencias
5 = Causa Falla

Padrón: 0

Propiedades: CFG, Net

Grupos de acceso vía HMI:

01 GRUPOS PARÁMETROS
∟ 49 Comunicación
∟ 111 Estados/Comandos

Descripción:
Este parámetro permite seleccionar cual acción debe ver ejecutada por el convertidor, caso un error de comunicación sea detectado.

Tabla 3.1 - *Valores para el parámetro P0313*

Opciones	Descripción
0 = Inactivo	Ninguna acción es ejecutada, el convertidor permanece en el estado actual.
1 = Para por Rampa	El comando de parada por rampa es ejecutado, y el motor para de acuerdo con la rampa de desaceleración programada.
2 = Deshabilita General	El convertidor es deshabilitado general, y el motor pera por inercia.
3 = Va para Modo Local	El convertidor es comandado para el modo local.
4 = Va para Modo Local y mantenga comandos y referencias	El convertidor de frecuencia es comandado para el modo local, más los comandos de habilita y de referencia (consigna) de velocidad recibidos vía red son mantenidos en modo local; caso el convertidor de frecuencia sea programado para utilizar en modo local comandos vía HMI o 3 wire start stop y referencia de velocidad vía HMI o potenciómetro electrónico.
5 = Causa Falla	En vez causar una alarma, un error de comunicación causará una falla en el convertidor de frecuencia. De este modo, será necesario hacer el reset de fallas para que el convertidor de frecuencia vuelva a su modo normal de operación.

Para la interface serial, es considerado error de comunicación solo el evento de *timeout* de la interface serial – alarma A128/falla F228. Este *timeout* es programado a través del parámetro P0314.

Las acciones descriptas en este parámetro son ejecutadas a través de la escrita automática de los respectivos bits en el parámetro de control vía serial / USB – P0682. Para que la acción ejecutada tenga efecto, es necesario que el convertidor se encuentre programado para ser controlado vía serial. Esta programación es hecha a través de los parámetros P0220 hasta P0228.

P0314 – Watchdog Serial

Rango: 0,0 … 999,0 s

Padrón: 0,0 s

Propiedades: CFG, Serial

Grupos de acceso vía HMI:

01 GRUPOS PARÁMETROS
∟ 49 Comunicación
∟ 113 Serial RS232/485

Descripción:
Permite programar un tiempo para la detección de error de comunicación vía interface serial. Caso el convertidor se quede sin recibir telegramas válidos por un tiempo mayor del que el programado en este parámetro, será considerado que ocurrió un error de comunicación, señalizando la alarma A128 en el HMI (o falla F228, dependiendo de la programación hecha en el parámetro P0313) y la acción programada en el P0313 será ejecutada.

Luego de energizado, el convertidor empezará a contar este tiempo a partir del primer telegrama válido recibido. El valor 0,0 deshabilita esta función.

P0316 – Estado de la Interface Serial

Rango: 0 = Inactivo
1 = Activo
2 = Error de *Watchdog*

Padrón: -

Propiedades: RO

Grupos de acceso vía HMI:

01 GRUPOS PARÁMETROS
∟ 49 Comunicación
∟ 113 Serial RS232/485

Descripción:
Permite identificar si la tarjeta de interface serial RS232 o RS485 está debidamente instalada, y si la comunicación serial presenta errores.

Tabla 3.2 *- Valores del parámetro P0316*

Opciones	Descripción
0 = Inactivo	Interface serial inactiva. Ocurre cuando el convertidor no posee tarjeta de interface RS232/RS485 instalado.
1 = Activo	Tarjeta de interface RS232/RS485 instalada y reconocida.
2 = Error de *Watchdog*	Interface serial activa, más detectado error de comunicación serial – alarme A128/falla F228.

P0680 – Estado Lógico

Rango: 0000h … FFFFh

Padrón: -

Propiedades: RO

Grupos de acceso vía HMI:

01 GRUPOS PARÁMETROS
∟ 49 Comunicación
∟ 111 Estados/Comandos

Descripción:
Permite el monitoreo del estado del convertidor. Cada bit representa un estado:

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3 a 0
Función	En Falla	Manual/ Automático	Subtensión	LOC/REM	JOG	Sentido de Giro	Habilitado General	Rampa Habilitada	En Alarma	En modo de configuración	Segunda Rampa	Parada Rápida Activada	Reservado

Tabla 3.3 *- Funciones de los bits para el parámetro P0680*

Bits	Valores
Bits 0 a 4	Reservado.
Bit 4 Parada Rápida Activada	**0:** Convertidor no possui comando de parada rápida activado. **1:** Convertidor ejecutando comando de parada rápida.
Bit 5 Segunda Rampa	**0:** Convertidor de frecuencia configurado para utilizar la primera rampa para acelerar y desacelerar el eje del motor; tiempos programados en los parámetros P0100 y P0101. **1:** Convertidor de frecuencia configurado para utilizar la segunda rampa para acelerar y desacelerar el eje del motor; tiempos programados en los parámetros P0102 y P0103.
Bit 6 En Modo de Configuración	**0:** Convertidor operando normalmente. **1:** Convertidor en modo de configuración. Indica una condición especial en la cual el convertidor no puede ser habilitado: ☑ Ejecutando rutina de autoajuste. ☑ Ejecutando rutina de puesta en marcha orientada. ☑ Ejecutando función *copy* de la HMI. ☑ Ejecutando rutina autodirigida de la tarjeta de memoria flash. ☑ Posee incompatibilidad de parametrización; ☑ Sin alimentación en el circuito de potencia del convertidor de frecuencia. **Obs.:** Es posible obtener la descripción exacta del modo especial de operación en el parámetro P0692.
Bit 7 En Alarma	**0:** Convertidor no está en el estado de alarma. **1:** Convertidor está en el estado de alarma. **Obs.:** el número de la alarma puede ser leído a través del parámetro P0048 – Alarma Actual.
Bit 8 Rampa Habilitada (RUN)	**0:** Motor está parado. **1:** Convertidor está girando el motor a la velocidad de referencia, o ejecutando rampa de aceleración o desaceleración.
Bit 9 Habilitado General	**0:** Convertidor está deshabilitado general. **1:** Convertidor está habilitado general y listo para girar el motor.
Bit 10 Sentido de Giro	**0:** Motor girando en el sentido reverso. **1:** Motor girando en el sentido directo.
Bit 11 JOG	**0:** Función JOG inactiva. **1:** Función JOG activa.
Bit 12 LOC/REM	**0:** Convertidor en el modo local. **1:** Convertidor en el modo remoto.
Bit 13 Subtensión	**0:** Sin subtensión. **1:** Con subtensión.
Bit 14 Manual/ Automático	**0:** En modo manual (función PID). **1:** En modo automático (función PID).
Bit 15 En Falla	**0:** Convertidor no está en el estado de falla. **1:** Alguna falla registrada por el convertidor. **Obs.:** El número de la falla puede ser leída a través del parámetro P0049 – Falla Actual.

P0681 – Velocidad del Motor en 13 bits

Rango: -32768 ... 32767 **Padrón:** -

Propiedades: RO

Grupos de acceso vía HMI:

01 GRUPOS PARÁMETROS
∟ 49 Comunicación
∟ 111 Estados/Comandos

Descripción:
Permite monitorear la velocidad del motor. Esta palabra utiliza resolución de 13 bits con señal para representar la rotación sincrónica del motor:

☑ P0681 = 0000h (0 decimal) → velocidad del motor = 0 rpm
☑ P0681 = 2000h (8192 decimal) → velocidad del motor = rotación sincrónica

Valores de velocidad en rpm intermediarios o superiores pueden ser obtenidos utilizando esta escala. Por ejemplo, para un motor de 4 polos y 1800 rpm de rotación sincrónica, caso el valor leído sea 2048 (0800h), para obtener el valor en rpm se debe calcular:

8192 – 1800 rpm
2048 – velocidad en rpm

$$\text{velocidad en rpm} = \frac{1800 \times 2048}{8192}$$

Velocidad en rpm = 450 rpm

Valores negativos para este parámetro indican motor girando en el sentido antihorario.

P0682 – Palabra de Control vía Serial / USB

Rango: 0000h … FFFFh **Padrón:** 0000h

Propiedades: Serial, USB

Grupos de acceso vía HMI:

01 GRUPOS PARÁMETROS
- 49 Comunicación
  - 111 Estados/Comandos

Descripción:
Palabra de comando del convertidor vía interface serial. Este parámetro solamente puede ser modificado vía interface serial o USB. Para las demás fuentes (HMI, CAN, etc.) se comporta como un parámetro solamente de lectura.

Para que los comandos escritos en este parámetro sean ejecutados, es necesario que el convertidor se encuentre programado para ser controlado vía serial. Esta programación es hecha a través de los parámetros P0105 y P0220 hasta P0228.

Cada bit de esta palabra representa un comando que puede ser ejecutado en el convertidor.

Bits	15 a 8	7	6	5	4	3	2	1	0
Función	Reservado	Reset de Fallas	Parada Rápida	Utiliza Segunda Rampa	LOC/REM	JOG	Sentido de Giro	Habilita General	Gira/Para

Tabla 3.4 *- Funciones de los bits para el parámetro P0682*

Bits	Valores
Bit 0 Gira/Para	**0:** Para motor por rampa de desaceleración. **1:** Gira motor de acuerdo con la rampa de aceleración hasta alcanzar el valor de la referencia de velocidad.
Bit 1 Habilita General	**0:** Deshabilita general el convertidor, interrumpiendo la alimentación para el motor. **1:** Habilita general el convertidor, permitiendo la operación del motor.
Bit 2 Sentido de Giro	**0:** Girar motor en el sentido opuesto al de la referencia. **1:** Girar motor en el sentido indicado en la referencia.
Bit 3 JOG	**0:** Deshabilita la función JOG. **1:** Habilita la función JOG.
Bit 4 LOC/REM	**0:** Convertidor va para el modo local. **1:** Convertidor va para el modo remoto.
Bit 5 Utiliza Segunda Rampa	**0:** Convertidor de frecuencia configurado para utiliza la primera rampa para acelerar y desacelerar el eje del motor, de acuerdo con los valores de tiempo programados en los parámetros P0100 y P0101. **1:** Convertidor de frecuencia configurado para utiliza la segunda rampa para acelerar y desacelerar el eje del motor, de acuerdo con los valores de tiempo programados en los parámetros P0102 y P0103.
Bit 6 Parada Rápida	**0:** No ejecuta comando de parada rápida. **1:** Ejecuta comando de parada rápida. **Obs.:** cuando el modo de control V/f o VVW está seleccionado, no se recomienda la utilización de esta función.
Bit 7 Reset de Fallas	**0:** Sin función. **1:** Si en estado de falla, ejecuta el reset del convertidor.
Bits 8 a 15	Reservado.

P0683 – Consigna de Velocidad vía Serial/ USB

Rango: -32768 … 32767 Padrón: 0

Propiedades: Serial, USB

Grupos de acceso vía HMI:

01 GRUPOS PARÁMETROS
∟ 49 Comunicación
∟ 111 Estados/Comandos

Descripción:
Permite programar la referencia de velocidad para el convertidor vía interface serial. Este parámetro solamente puede ser modificado vía interface serial o USB. Para las demás fuentes (HMI, CAN, etc.) se comporta como un parámetro solamente de lectura.

Para que la referencia escrita en este parámetro sea utilizada, es necesario que el convertidor se encuentre programado para utilizar la consigna de velocidad vía serial. Esta programación es hecha a través de los parámetros P0221 y P0222.

Esta palabra utiliza resolución de 13 bits con señal para representar la rotación sincrónica del motor:

☑ P0683 = 0000h (0 decimal) → referencia de velocidad = 0 rpm
☑ P0683 = 2000h (8192 decimal) → referencia de velocidad = rotación sincrónica

Valores de referencia intermediarios o superiores pueden ser programados utilizando esta escala. Por ejemplo, para un motor de 4 polos y 1800 rpm de rotación sincrónica, caso se desee una referencia de 900 rpm, se debe calcular:

1800 rpm – 8192
900 rpm – referencia en 13 bit

$$\text{referencia en 13 bit} = \frac{900 \times 8192}{1800}$$

Referencia en 13 bit = 4096 (valor correspondiente a 900 rpm en la escala en 13 bits)

Este parámetro también acepta valores negativos para cambiar el sentido de rotación del eje el motor. El sentido de rotación de la referencia, sin embargo, depende también del valor del bit 2 de la palabra de control – P0682:

☑ Bit 2 = 1 y P0683 > 0: referencia para el sentido directo
☑ Bit 2 = 1 y P0683 < 0: referencia para el sentido reverso
☑ Bit 2 = 0 y P0683 > 0: referencia para el sentido reverso
☑ Bit 2 = 0 y P0683 < 0: referencia para el sentido directo

P0695 – Valor para las Salidas Digitales

Rango: 0000h … FFFFh **Padrón:** 0000h

Propiedades: Net

Grupos de acceso vía HMI:

01 GRUPOS PARÁMETROS
∟ 49 Comunicación
∟ 111 Estados/Comandos

Descripción:
Posibilita el control de las salidas digitales a través de las interfaces de red (Serial, USB, CAN, etc.). Este parámetro no puede ser modificado a través de la HMI.

Cada bit de este parámetro corresponde al valor deseado para una salida digital. Para que la salida digital correspondiente pueda ser controlada de acuerdo con este contenido, se hace necesario que su función sea programada para "Contenido P0695", en los parámetros P0275 a P0280.

Bits	15 a 5	4	3	2	1	0
Función	Reservado	Valor para DO5	Valor para DO4	Valor para DO3 (RL3)	Valor para DO2 (RL2)	Valor para DO1 (RL1)

Tabla 3.5 *- Funciones de los bits para el parámetro P0695*

Bits	Valores
Bit 0 Valor para DO1 (RL1)	**0:** salida DO1 abierta. **1:** salida DO1 cerrada.
Bit 1 Valor para DO2 (RL2)	**0:** salida DO2 abierta. **1:** salida DO2 cerrada.
Bit 2 Valor para DO3 (RL3)	**0:** salida DO3 abierta. **1:** salida DO3 cerrada.
Bit 3 Valor para DO4	**0:** salida DO4 abierta. **1:** salida DO4 cerrada.
Bit 4 Valor para DO5	**0:** salida DO5 abierta. **1:** salida DO5 cerrada.
Bits 5 a 15	Reservado

P0696 – Valor 1 para Salidas Analógicas
P0697 – Valor 2 para Salidas Analógicas
P0698 – Valor 3 para Salidas Analógicas
P0699 – Valor 4 para Salidas Analógicas

Rango: -32768 … 32767 **Padrón:** 0

Propiedades: Net

Grupos de acceso vía HMI:

01 GRUPOS PARÁMETROS
∟ 49 Comunicación
∟ 111 Estados/Comandos

Descripción:
Posibilita el control de las salidas analógicas a través de las interfaces de red (Serial, USB, CAN, etc.). Este parámetro no puede ser modificado a través de la HMI.

El valor escrito en estos parámetros es utilizado como valor para la salida analógica, desde que la función de la salida analógica deseada sea programada para "Contenido P0696/P0697/ P0698/ P0699", en los parámetros P0251, P0254, P0257 o P0260.

El valor debe ser escrito en una escala de 15 bits (7FFFh = 32767)[2] para representar 100% del valor deseado para la salida, o sea:

☑ P0696 = 0000h (0 decimal) → valor para la salida analógica = 0 %
☑ P0696 = 7FFFh (32767 decimal) → valor para la salida analógica = 100 %

En este ejemplo fue presentado el parámetro P0696, más la misma escala es utilizada para los parámetros P0697/P0698/P0699. Por ejemplo, se quiere controlar el valor de la salida analógica 1 a través da serial. En este caso se debe hacer la siguiente programación:

☑ Elegir un de los parámetros P0696 a P0699 para ser el valor utilizado por la salida analógica 1. En este ejemplo, vamos elegir el P0696.
☑ Programar, en la función de la salida analógica 1 (P0254), la opción "Contenido P0696".
☑ A través de la interface serial, escribir en el P0696 el valor deseado para la salida analógica 1, entre 0 e 100 %, de acuerdo con la escala del parámetro.

[2] Para la resolución real de la salida, consulte el manual del CFW-11.

¡NOTA!

Caso la salida analógica sea programada para operar de -10 V hasta 10 V, valores negativos para estos parámetros deben ser utilizados para comandar la salida con valores negativos de tensión, o sea, -32769 hasta 32767 representa una variación de -10 V hasta 10 V en la salida analógica.

4 Protocolo TP

El protocolo TP fue desarrollado con el objetivo de posibilitar la comunicación con PLCs de la línea TP. Más debido a su flexibilidad y facilidad de uso, también ha sido utilizado en otras aplicaciones, siendo muchas veces implementado en PLCs y otros sistemas para el control y el monitoreo de los equipos WEG.

4.1 Campos do Protocolo

- ☑ **STX:** Byte de "Start of Transmition": Valor: 02h; 2 decimal.
- ☑ **ETX:** Byte de "End of Transmition": Valor: 03h; 3 decimal.
- ☑ **ADR:** Byte de la dirección del convertidor en la red, programable a través del P0308.
  Rango de Valores: 41h; 65 decimal; 'A' (ASCII) até 5Eh; 94 decimal; '^' (ASCII), representando las direcciones de 1... 30 en la red.
  Especial 1: 40h; 64 decimal; '@' (ASCII) → Permite escrita o lectura de todos los equipos conectados a la red.
  Especial 2: 5Fh; 95 decimal; '_' (ASCII) → Permite SOLAMENTE escrita en todos los equipos conectados a la red sin respuesta de aceptación o rechazamiento.

- ☑ **COD:** Byte del Código del Telegrama
  Lectura: 3Ch (hexadecimal); 60 (decimal); '<' (ASCII)
  Escrita: 3Dh (hexadecimal); 61 (decimal); '=' (ASCII) sin guardar el parámetro en la EEPROM
  Escrita: 3Eh (hexadecimal); 62 (decimal); '>' (ASCII) guardando el parámetro en la EEPROM

- ☑ **BCC:** Byte de Checksum longitudinal del telegrama, O EXCLUSIVO (XOR) entre todos los bytes del telegrama. Tamaño de 1 byte (00h ... FFh hexadecimal)

- ☑ **DMW:** "Data Master Write". Son 4 bytes de escrita que el maestro envía al esclavo, siendo que los 2 primeros representan el parámetro y los 2 últimos el valor a ser escrito en este parámetro.
  PHi: Byte representando la parte alta del parámetro
  PLo: Byte representando la parte baja del parámetro
  VHi: Byte representando la parte alta del valor a ser escrito
  VLo: Byte representando la parte baja del valor a ser escrito
  Ejemplo: Escribir 1FFFh en la consigna de velocidad (P0683) → PHi = 02h, PLo = ABh, VHi = 1Fh, VLo = FFh.

- ☑ **DMR:** "Data Master Read". Son 2 bytes de lectura que el maestro envía al esclavo que representan el parámetro a ser leído.
  PHi: Byte representando la parte alta del parámetro
  PLo: Byte representando la parte baja del parámetro
  Ejemplo: Leer el valor contenido en el parámetro de tensión de salida (P0007) → PHi = 00h, PLo = 07h.

- ☑ **NUM:** Byte que representa el número de DMW o DMR a sierren transmitidos, conforme el COD del telegrama.
  Rango de valores: 1 ... 6 (decimal).

- ☑ **DSV:** "Data Slave Value". Son 2 bytes que el esclavo envía al maestro luego de una solicitación de un telegrama de lectura del maestro, representando el valor contenido en el parámetro solicitado.
  VHi: Byte representando la parte alta del valor a ser escrito
  VLo: Byte representando la parte baja del valor a ser escrito
  Ejemplo: Respuesta a la solicitación de lectura del parámetro de estado del convertidor (P0680) → VHi = 13h, VLo = 00h.

- ☑ **ACK:** Byte de aceptación del esclavo luego de una escrita del maestro. Valor: 06h; 6 decimal;

- ☑ **NAK:** Byte de rechazamiento del esclavo luego de una lectura o escrita del maestro. Puede ocurrir cuando el maestro solicita una escrita o lectura de un parámetro inexistente o el valor a ser escrito en el parámetro está fuera del rango de valores permitido.
  Valor: 15h; 21 decimal;

4.2 Formato de los Telegramas

A seguir serán presentados los formatos de los telegramas de lectura y escrita en parámetros. Es importante observar que cada telegrama en el protocolo TP permite realizar la lectura o escrita de hasta 6 parámetros por vez. Telegramas que poseen error en el formato o BCC incorrecto serán ignorados por el convertidor, que no enviará respuesta para el maestro.

4.2.1 Telegrama de Lectura

Maestro:

STX	ADR	COD	NUM	**DMR**	...	**DMR**	ETX	BCC

- ☑ COD: código para lectura →3Ch (hexadecimal); 60 (decimal); '<' (ASCII)
- ☑ NUM: número de parámetros leídos. Rango de 1 ... 6.
- ☑ DMR: número del parámetro solicitado. El número de DMRs debe ser igual al valor configurado en el byte NUM.

Esclavo (CFW-11):

ADR	**DSV**	...	**DSV**	BCC

o

ADR	NAK

- ☑ DSV: valor del parámetro solicitado. El número de DSVs es igual al valor configurado en el byte NUM

Recordando que:

DMR	
PHi	PLo

DSV	
VHi	VLo

4.2.2 Telegrama de Escrita

Maestro:

STX	ADR	COD	NUM	**DMW**	...	**DMW**	ETX	BCC

- ☑ COD: código para escrita
  - 3Eh (hexadecimal); 62 (decimal); '>' (ASCII) → guardando en la EEPROM
  - 3Dh (hexadecimal); 61 (decimal); '=' (ASCII) → sin guardar en la EEPROM
- ☑ NUM: número de parámetros escritos. Rango de 1 ... 6.
- ☑ DMW: número y contenido para el parámetro. El número de DMWs debe ser igual al valor configurado en el byte NUM.

Esclavo (CFW-11):

ADR	ACK

o

ADR	NAK

Recordando que:

DMW			
PHi	PLo	VHi	VLo

4.3 Ejemplo de Telegramas Utilizando el Protocolo TP

Todos ejemplos a seguir consideran que el convertidor está programado con la dirección 1 (P0308=1), luego el campo ADR es enviado con valor 41h (consulte Tabla I.1).

Ejemplo 1: lectura de dos parámetros del convertidor:
- ☑ Velocidad del motor – P0002 (suponiendo P0002 en 1200 rpm = 04B0h).
- ☑ Corriente del motor – P0003 (suponiendo P0003 en 5,0A = 0032h).

Maestro:

02h	41h	3Ch	02h	00h	02h	00h	03h	03h	7Fh
STX	ADR	COD	NUM	DMR:P0002		DMR:P0003		ETX	BCC
				Parámetro		Parámetro			

Esclavo (CFW-11):

41h	04h	B0h	00h	32h	C7h
ADR	DSV:1200		DSV:50		BCC
	Valor		Valor		

Ejemplo 2: Programar 6 parámetros para la operación del convertidor:
- ☑ Telegrama de escrita guardado en la EEPROM.
- ☑ P0100 = 50 (100 en decimal = 0064h, 50 en decimal = 0032h)
- ☑ P0101 = 150 (101 en decimal = 0065h, 150 en decimal = 0096h)
- ☑ P0220 = 6 (220 en decimal = 00DChh, 6 en decimal = 0006h)
- ☑ P0222 = 9 (222 en decimal = 00DEh, 9 en decimal = 0009h)
- ☑ P0226 = 5 (226 en decimal = 00E2h, 5 en decimal = 0005h)
- ☑ P0227 = 2 (227 en decimal = 00E3h, 2 en decimal = 0002h)

Maestro (solicitando):

02h	41h	3Eh	06h	00h	64h	00h	32h	00h	65h	00h	96h
STX	ADR	COD	NUM	DMW:P0100 = 50				DMW:P0101 = 150			
				Parámetro		Valor		Parámetro		Valor	

00h	DCh	00h	06h	00h	DEh	00h	09h	00h	E2h	00h	05h
DMW:P0220 = 6				DMW:P0222 = 9				DMW:P0226 = 5			
Parámetro		Valor		Parámetro		Valor		Parámetro		Valor	

00h	E3h	00h	02h	03h	D6h
DMW:P0227 = 2				ETX	BCC
Parámetro		Valor			

Esclavo (respuesta):

41h	06h
ADR	ACK

Ejemplo 4: escrita del comando de habilitar y de la consigna de velocidad vía serial:
- ☑ Telegrama de escrita sin guardar en la EEPROM.
- ☑ P0682 = 0013h – control vía serial con comandos de modificación para modo remoto, habilita general y habilita rampa (682 en decimal = 02AAh).
- ☑ P0683 = 1000h – consigna de velocidad vía serial programada para la mitad de la velocidad sincrónica del motor (683 en decimal = 02ABh).

Maestro (solicitando):

02h	41h	3Dh	02h	02h	AAh	00h	13h	02h	ABh	10h	00h	03h	7Dh
STX	ADR	COD	NUM	DMW:0682 = 0013h				DMW:P0683 = 1000h				ETX	BCC
				Parámetro		Valor		Parámetro		Valor			

Esclavo (respuesta):

41h	06h
ADR	ACK

5 Protocolo Modbus-RTU

El protocolo Modbus fue inicialmente desarrollado en 1979. Actualmente, es un protocolo abierto ampliamente difundido, utilizado por varios fabricantes en diversos equipamientos. La comunicación Modbus-RTU del convertidor CFW-11 fue desarrollada con base en los siguientes documentos:

- ☑ MODBUS Protocol Reference Guide Rev. J, MODICON, June 1996.
- ☑ MODBUS Application Protocol Specification, MODBUS.ORG, May 8th 2002.
- ☑ MODBUS over Serial Line, MODBUS.ORG, December 2nd 2002.

En estos documentos están definidos los formatos de los mensajes utilizados por los elementos que hacen parte de la red Modbus, los servicios (o funciones) que pueden ser estar disponibles vía red, y también como estos elementos intercambian datos en la red.

5.1 Modos de Transmisión

En la especificación del protocolo están definidos dos modos de transmisión: ASCII y RTU. Los modos definen la forma como son transmitidos los bytes del mensaje No es posible utilizar los dos modos de transmisión en la misma red.

El convertidor de frecuencia CFW-11 utiliza solamente el modo RTU para la transmisión de telegramas. Los bytes son transmitidos en el formato hexadecimal, y su configuración depende de la programación hecha a través del P0311.

5.2 Estructura de los Mensajes en el Modo RTU

La red Modbus-RTU utiliza el sistema maestro-esclavo para el intercambio de mensajes. Permite hasta 247 esclavos, más solamente un maestro. Toda comunicación inicia con el maestro haciendo una solicitación a un esclavo, y este contesta al maestro el que fue solicitado. En ambos los telegramas (pregunta y respuesta), la estructura utilizada es la misma: Dirección, Código de la Función, Datos y CRC. Luego el campo de datos podrá tener tamaño variable, dependiendo del que está siendo solicitado.

Maestro (telegrama de solicitación):

Dirección (1 byte)	Función (1 byte)	Datos de la solicitación (n bytes)	CRC (2 bytes)

Esclavo (telegrama de respuesta):

Dirección (1 byte)	Función (1 byte)	Datos de la respuesta (n bytes)	CRC (2 bytes)

5.2.1 Dirección

El maestro inicia la comunicación enviando un byte con la dirección del esclavo para el cual se destina el mensaje. Al enviar la respuesta, el esclavo también inicia el telegrama con el su propia dirección. El maestro también puede enviar un mensaje destinado a la dirección "0" (cero), lo que significa que el mensaje es destinada a todos los esclavos de la red (*broadcast*). En este caso, ninguno esclavo irá contestar al maestro.

5.2.2 Código de la Función

Este campo también contiene un único byte, donde el maestro especifica el tipo de servicio o función solicitada al esclavo (lectura, escrita, etc.). De acuerdo con el protocolo, cada función es utilizada para acceder un tipo específico de dato.

En el CFW-11, los datos relativos a los parámetros están disponibles como registradores del tipo *holding* (referenciados a partir de la dirección 40000 o '4*x*').

5.2.3 Campo de Datos

Campo con tamaño variable. El formato y el contenido de este campo dependen de la función utilizada y de los valores transmitidos. Este campo está descrito juntamente con la descripción de las funciones (consultar ítem 5.4).

5.2.4 CRC

La última parte del telegrama es el campo para el chequeo de errores de transmisión. El método utilizado es el CRC-16 (Cycling Redundancy Check). Este campo es formado por dos bytes, donde primero es transmitido el byte menos significativo (CRC-), y después el más significativo (CRC+). La forma de cálculo del CRC es descrita en la especificación del protocolo, sin embargo informaciones para su implementación también son suministradas en los apéndices B y C.

5.2.5 Tiempo entre Mensajes

En el modo RTU no existe un carácter específico que indique el inicio o el fin de un telegrama. La indicación de cuando un nuevo mensaje empieza o cuando elle termina es hecha por la ausencia de transmisión de datos en la red, por un tiempo mínimo de 3,5 veces el tiempo de transmisión de un byte de datos (11 bits[3]). Siendo así, caso un telegrama tenga iniciado luego de transcurrido este tiempo mínimo, los elementos de la red irán asumir que el primero carácter recibido representa el inicio de un nuevo telegrama. Y de la misma forma, los elementos de la red irán asumir que el telegrama ha llegado al fin cuando, recibidos los bytes del telegrama, este tiempo transcurrir nuevamente.

Si durante la transmisión de un telegrama, el tempo entre los bytes fue mayor que este tempo mínimo, el telegrama será considerado inválido, pues el convertidor irá rechazar los bytes ya recibidos y montará un nuevo telegrama con los bytes que estuvieren siendo transmitidos.

Para tasas de comunicación superiores a 19200 bits/s, los tiempos utilizados son los mismos que para esta tasa. La tabla a seguir preséntanos los tiempos para distintas tasas de comunicación:

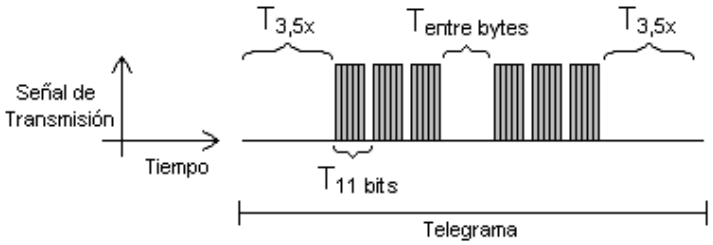


Tabla 5.1 *- Tasas de comunicación y tiempos involucrados en la transmisión de telegramas*

Tasa de Comunicación	$T_{11\ bits}$	$T_{3.5x}$
9600 bits/s	1,146 ms	4,010 ms
19200 bits/s	573 µs	2,005 ms
38400 bits/s	573 µs	2,005 ms
57600 bits/s	573 µs	2,005 ms

- ☑ $T_{11\ bits}$ = Tiempo para transmitir una palabra del telegrama.
- ☑ $T_{entre\ bytes}$ = Tiempo entre bytes (no puede ser mayor que $T_{3,5x}$).
- ☑ $T_{3,5x}$ = Intervalo mínimo para indicar el inicio y el fin de telegrama (3,5 x T_{11bits}).

[3] Siempre es considerado el tiempo de 11 bits como el tiempo para transmisión de un byte, mismo que en el parámetro P0311 sea programado un formato de telegrama donde cada byte posea solo 10 bits.

5.3 Operación del CFW-11 en la Red MODBUS-RTU

El CFW-11 posee las siguientes características cuando operado en red Modbus-RTU:

☑ Conexión de la red vía interface serial RS-232 o RS-485 (consulte ítem 2).
☑ La dirección, tasa de comunicación y formato de los bytes definidos a través de parámetros (consulte ítem 3).
☑ Permite la parametrización y control del convertidor a través del acceso a parámetros.

5.3.1 Funciones Disponibles y Tiempos de Respuesta

En la especificación del protocolo Modbus-RTU son definidas funciones utilizadas para acceder diferentes tipos de registradores. En el CFW-11, los parámetros fueran definidos como siendo registradores del tipo *holding*. Para acceder estos registradores, fueran colocados disponibles los siguientes servicios (o funciones):

☑ *Read Coils*[4]
Descripción: lectura de bloque bits del tipo *coil*.
Código de la función: 01.

☑ *Read Discrete Inputs*[4]
Descripción: lectura de bloque bits del tipo entradas discretas.
Código de la función: 02.

☑ *Read Holding Registers*
Descripción: lectura de bloque de registradores del tipo *holding*.
Código de la función: 03.

☑ *Read Input Registers*[4]
Descripción: lectura de bloque de registradores del tipo *input*.
Código de la función: 04.

☑ *Write Single Coil*[4]
Descripción: escrita en un único bit del tipo *coil*.
Código de la función: 05.

☑ *Write Single Register*
Descripción: escrita en un único registrador del tipo *holding*.
Código de la función: 06.

☑ *Write Multiple Coils*[4]
Descripción: escrita en bloque de bit del tipo *coil*.
Código de la función: 15.

☑ *Write Multiple Registers*
Descripción: escrita en bloque de registradores del tipo *holding*.
Código de la función: 16.

☑ *Read Device Identification*
Descripción: identificación del modelo del drive.
Código de la función: 43.

El tiempo de respuesta del convertidor CFW-11, fin de la transmisión del maestro hasta el inicio de la respuesta del esclavo, varia de 2 a 10 ms, para cualquier una de las funciones arriba

[4] Funciones utilizadas para acceder a los datos utilizados por la función SoftPLC

5.3.2 Dirección de los Datos y Offset

La dirección de los datos en el CFW-11 es hecha con offset igual a cero, lo que significa que el número de la dirección equivale al número dado. Los parámetros están disponibles a partir de la dirección "0" (cero). La tabla que sigue presenta la dirección de los parámetros, que pueden ser accedidos como registradores del tipo *holding*:

Tabla 5.2 *- Dirección de los datos para la interface Modbus-RTU*

Parámetros		
Número del Parámetro	**Dirección del dato Modbus**	
	Decimal	**Hexadecimal**
P0000	0	0000h
P0001	1	0001h
⋮	⋮	⋮
P0100	100	0064h
⋮	⋮	⋮

¡NOTA!

☑ Todos los parámetros son tratados como registradores del tipo holding. Dependiendo del maestro utilizado, estos registradores son referenciados a partir del enderezo base 40000 o 4x. En este caso, la dirección para un parámetro que debe ser programado en el maestro es la dirección presentada en la tabla arriba adicionado a la dirección base. Consulte la documentación del maestro para saber como acceder registradores del tipo *holding*.

☑ Además de los parámetros, otros tipos de datos como marcadores de bit, *word* o *float* también pueden ser accedidos utilizando la interface Modbus-RTU. Estos marcadores son utilizados principalmente por la función SoftPLC disponible para el CFW-11. Para la descripción de estos marcadores, bien como la dirección para accederlos vía Modbus, se debe consultar el Manual de la SoftPLC.

5.4 Descripción Detallada de las Funciones

En este ítem es hecha una descripción detallada de las funciones disponibles en el CFW-11 para comunicación Modbus-RTU. Para la elaboración de los telegramas, es importante observar lo siguiente:

☑ Los valores son siempre transmitidos en hexadecimal.

☑ La dirección de un dato, el número de datos y el valor de los registradores son siempre representados en 16 bits. Por eso, es necesario transmitir estos campos utilizando dos bytes – superior (*high*) e inferior (*low*).

☑ Los telegramas, tanto para pregunta cuanto para respuesta, no pueden ultrapasar 64 bytes.

☑ Los valores transmitidos son siempre números enteros, independiente de poseyeren representación con casa decimal. De esta forma, el valor 9,5 seria transmitido como siendo 95 (5Fh) vía serial. Consulte la lista de parámetro del CFW-11 para obtener la resolución utilizada para cada parámetro.

5.4.1 Función 03 – Read Holding Register

Lee el contenido de un grupo de registradores, que necesariamente deben estar en secuencia numérica. Esta función posee la siguiente estructura para los telegramas de lectura y respuesta (los valores sin siempre hexadecimal, y cada campo representa un byte):

Pregunta (Maestro)	**Respuesta (Esclavo)**
Dirección del esclavo	Dirección del esclavo
Función	Función
Dirección del registrador inicial (byte high)	Campo Byte Count
Dirección del registrador inicial (byte low)	Dato 1 (high)
Número de registradores (byte high)	Dato 1 (low)
Número de registradores (byte low)	Dato 2 (high)
CRC-	Dato 2 (low)
CRC+	Etc...
	CRC-
	CRC+

Ejemplo 1: lectura de la velocidad del motor (P0002) y corriente del motor (P0003) del CFW-11 en la dirección 1 (suponiendo P0002 = 1000 rpm y P0003 = 3,5 A).

☑ Dirección: 1 = 01h (1 byte)
☑ Número del primer parámetro: 2 = 0002h (2 bytes)
☑ Valor del primer parámetro: 1000 = 03E8h (2 bytes)
☑ Valor del segundo parámetro: 35 = 0023h (2 bytes)

Pregunta (Maestro)		**Respuesta (Esclavo)**	
Campo	***Valor***	***Campo***	***Valor***
Dirección del esclavo	01h	Dirección del esclavo	01h
Función	03h	Función	03h
Registrador inicial (high)	00h	Byte Count	04h
Registrador inicial (low)	02h	P002 (high)	03h
No. de registradores (high)	00h	P002 (low)	E8h
No. de registradores (low)	02h	P003 (high)	00h
CRC-	65h	P003 (low)	23h
CRC+	CBh	CRC-	3Bh
		CRC+	9Ah

5.4.2 Función 06 – Write Single Register

Esta función es utilizada para escribir un valor para un único registrador. Posee la siguiente estructura (los valores son siempre hexadecimal, y cada campo representa un byte):

Pregunta (Maestro)	**Respuesta (Esclavo)**
Dirección del esclavo	Dirección del esclavo
Función	Función
Dirección del registrador (byte high)	Dirección del registrador (byte high)
Dirección del registrador (byte low)	Dirección del registrador (byte low)
Valor para el registrador (byte high)	Valor para el registrador (byte high)
Valor para el registrador (byte low)	Valor para el registrador (byte low)
CRC-	CRC-
CRC+	CRC+

Ejemplo 2: escrita de la consigna de velocidad (P0683) en 900 rpm (suponiendo velocidad sincrónica de 1800 rpm), para el CFW-11 en la dirección 3.

☑ Dirección: 3 = 03h (1 byte)
☑ Número del parámetro: 683 = 02AB (2 bytes)
☑ Valor para el parámetro: 1000h (2 bytes)

Pregunta (Maestro)		**Respuesta (Esclavo)**	
Campo	***Valor***	***Campo***	***Valor***
Dirección del esclavo	03h	Dirección del esclavo	03h
Función	06h	Función	06h
Registrador (high)	02h	Registrador (high)	02h
Registrador (low)	ABh	Registrador (low)	ABh
Valor (high)	10h	Valor (high)	10h
Valor (low)	00h	Valor (low)	00h
CRC-	F5h	CRC-	F5h
CRC+	B0h	CRC+	B0h

Note que para esta función, la respuesta del esclavo es una copia idéntica de la solicitación hecha por el maestro.

5.4.3 Función 16 – Write Multiple Registers

Esta función permite escribir valores para un grupo de registradores, que deben estar en secuencia numérica. También puede ser usada para escribir un único registrador (los valores son siempre hexadecimal, e cada campo representa un byte).

Pregunta (Maestro)	Respuesta (Esclavo)
Dirección del esclavo	Dirección del esclavo
Función	Función
Dirección del registrador inicial (byte high)	Dirección del registrador inicial (byte high)
Dirección del registrador inicial (byte low)	Dirección del registrador inicial (byte low)
Número de registradores (byte high)	Número de registradores (byte high)
Número de registradores (byte low)	Número de registradores (byte low)
Campo Byte Count (nº de bytes de datos)	CRC-
Dato 1 (high)	CRC+
Dato 1 (low)	
Dato 2 (high)	
Dato 2 (low)	
Etc...	
CRC-	
CRC+	

Ejemplo 3: escrita del tiempo de aceleración (P0100) igual a 1,0s y tiempo de desaceleración (P0101) igual a 2,0s, de un CFW-11 en la dirección 15.

☑ Valores convertidos para hexadecimal:
- Dirección: 15 = 0Fh (1 byte)
- Número del primer parámetro: 100 = 0064h (2 bytes)
- Valor para el primer parámetro: 10 = 000Ah (2 bytes)
- Valor para el segundo parámetro: 20 = 0014h (2 bytes)

Pregunta (Maestro)		Respuesta (Esclavo)	
Campo	***Valor***	***Campo***	***Valor***
Dirección del esclavo	0Fh	Dirección del esclavo	0Fh
Función	10h	Función	10h
Registrador inicial (high)	00h	Registrador (high)	00h
Registrador inicial (low)	64h	Registrador (low)	64h
No. de registradores (high)	00h	Valor (high)	00h
No. de registradores (low)	02h	Valor (low)	02h
Byte Count	04h	CRC-	01h
P100 (high)	00h	CRC+	39h
P100 (low)	0Ah		
P101 (high)	00h		
P101 (low)	14h		
CRC-	E0h		
CRC+	91h		

5.4.4 Función 43 – Read Device Identification

Función auxiliar, que permite la lectura del fabricante, modelo y versión de "firmware" del producto. Posee la siguiente estructura:

Pregunta (Maestro)	Respuesta (Esclavo)
Dirección del esclavo	Dirección del esclavo
Función	Función
MEI Type	MEI Type
Código de lectura	Conformity Level
Número del Objeto	More Follows
CRC-	Próximo objeto
CRC+	Número de objetos
	Código del primer objeto
	Tamaño del primer objeto
	Valor del primer objeto (n bytes)
	Código del segundo objeto
	Tamaño del segundo objeto
	Valor del segundo objeto (n bytes)
	etc...
	CRC-
	CRC+

Esta función permite la lectura de tres categorías de informaciones: Básica, Regular y Extendida, y cada categoría es formada por un grupo de objetos. Cada objeto es formado por una secuencia de caracteres ASCII. Para el CFW-11, solo informaciones básicas están disponibles, formadas por tres objetos:

- ☑ Objeto 00h – VendorName: Siempre 'WEG'.
- ☑ Objeto 01h – ProductCode: Formado por el código del producto (CFW-11) más la tensión y corriente nominal del convertidor (ex. 'CFW-11 220 - 230 V 10A/8A').
- ☑ Objeto 02h – MajorMinorRevision: indica la versión de firmware del drive, en el formato 'VX.XX'.

El código de lectura indica cuales las categorías de informaciones son leídas, y si los objetos son accedidos en secuencia o individualmente. En el caso, el CFW-11 soporta los códigos 01 (informaciones básicas en secuencia), y 04 (acceso individual a los objetos). Los demás campos son especificados por el protocolo y para el CFW-11 poseen valores fijos.

Ejemplo 4: lectura de las informaciones básicas en secuencia, a partir del objeto 01h, de un CFW-11 en la dirección 1:

Pregunta (Maestro)		**Respuesta (Esclavo)**	
Campo	***Valor***	***Campo***	***Valor***
Dirección del esclavo	01h	Dirección del esclavo	01h
Función	2Bh	Función	2Bh
MEI Type	0Eh	MEI Type	0Eh
Código de lectura	01h	Código de lectura	01h
Número del Objeto	01h	Conformity Level	81h
CRC-	70h	More Follows	00h
CRC+	77h	Próximo Objeto	00h
		Número de objetos	02h
		Código del Objeto	01h
		Tamaño del Objeto	1Bh
		Valor del Objeto	'CFW-11 220 - 230 V 10A / 8A'
		Código del Objeto	02h
		Tamaño del Objeto	05h
		Valor del Objeto	'V4.50'
		CRC-	B2h
		CRC+	8Fh

En este ejemplo, el valor de los objetos no fue representado en hexadecimal, más sí utilizando los caracteres ASCII correspondientes. Por ejemplo, para el objeto 02h, el valor 'V4.50' fue transmitido como siendo cinco caracteres ASCII, que en hexadecimal poseen los valores 56h ('V'), 34h ('4'), 2Eh ('.'), 35h ('5') y 30h ('0').

5.4.5 Errores de Comunicación

Errores de comunicación pueden ocurrir tanto en la transmisión de los telegramas cuanto en el contenido de los telegramas transmitidos. De acuerdo con el tipo de error, el CFW-11 podrá o no enviar respuesta para el maestro. Cuando el maestro envía un mensaje para un drive configurado en una determinada dirección de la red, el convertidor no irá contestar al maestro caso ocurra:

- ☑ Error en el bit de paridad.
- ☑ Error en el CRC.
- ☑ *Timeout* entre los bytes transmitidos (3,5 veces el tempo de transmisión de un byte).

En estos casos, el maestro deberá detectar la ocurrencia del error por el *timeout* en la espera de la contestación del esclavo. En el caso de una recepción con suceso, durante el tratamiento del telegrama, el convertidor puede detectar problemas y enviar un mensaje de error, indicando el tipo de problema encontrado:

- ☑ Función inválida (código del error = 1): la función solicitada no está implementada para el equipamiento.
- ☑ Dirección de dato inválido (código del error = 2): la dirección del dato (parámetro) no existe.
- ☑ Valor de dato inválido (código del error = 3): ocurre en las siguientes situaciones:
  - Valor está fuera del rango permitido.
  - Escrita en dato que no puede ser modificado (registrador solamente de lectura).

¡NOTA!
Es importante que sea posible identificar en el maestro cual el tipo de error ocurrido para que se pueda diagnosticar problemas durante la comunicación.

En el caso de la ocurrencia de algún de estos errores, el esclavo debe retornar un mensaje para el maestro que indica el tipo de error ocurrido. Los mensajes de error enviados por el esclavo poseen la siguiente estructura:

Pregunta (Maestro)	**Respuesta (Esclavo)**
Dirección del esclavo	Dirección del esclavo
Función	Función (con el bit más significativo en 1)
Datos	Código del error
CRC-	CRC-
CRC+	CRC+

Ejemplo 5: maestro solicita para el esclavo de la dirección 1 la escrita en el parámetro 99 (parámetro inexistente):

Pregunta (Maestro)		**Respuesta (Esclavo)**	
Campo	***Valor***	***Campo***	***Valor***
Dirección del esclavo	01h	Dirección del esclavo	01h
Función	06h	Función	86h
Registrador (high)	00h	Código de error	02h
Registrador (low)	63h	CRC-	C3h
Valor (high)	00h	CRC+	A1h
Valor (low)	00h		
CRC-	79h		
CRC+	D4h		

6 Fallas y Alarmas Relacionadas con la Comunicación Serial

A128/F228 – Timeout en la Recepción de Telegramas

Descripción:
Único alarma/falla relacionado con la comunicación serial. Indica que el convertidor ha parado de recibir telegramas seriales válidos por un período mayor del que el programado en el P0314.

Actuación:
El parámetro P0314 permite programar un tiempo dentro del cual el convertidor deberá recibir al menos un telegrama válido vía interface serial RS232/RS485 – con dirección y campo de chequeo de errores correctos – caso contrario será considerado que ha ocurrido algún problema en la comunicación serial. El conteo del tiempo es iniciado luego de la recepción del primer telegrama válido. Esta función puede ser utilizada para cualquier protocolo serial soportado por el convertidor.

Después de identificado el *timeout* en la comunicación serial, será señalizada a través del HMI el mensaje de alarma A128 – o falla F228, dependiendo de la programación hecha en el parámetro P0313. En condición de alarma, caso la comunicación sea restablecida y nuevos telegramas válidos sean recibidos, la señalización de falla en el IHM desaparecerá automáticamente.

Posibles Causas/Corrección:
- ☑ Verificar factores que puedan provocar fallas en la comunicación (cables, instalación, puesta a tierra).
- ☑ Garantizar que el maestro envíe telegramas para el convertidor siempre en un tiempo menor que el programado en el P0314.
- ☑ Deshabilitar esta función en el P0314.

I. APÉNDICES

Apéndice A. Tabla ASCII

Tabla I.1 - *Caracteres ASCII*

Dec	Hex	Chr		Dec	Hex	Chr	Dec	Hex	Chr	Dec	Hex	Chr
0	00	NUL	(Null char.)	32	20	Sp	64	40	@	96	60	`
1	01	SOH	(Start of Header)	33	21	!	65	41	A	97	61	a
2	02	STX	(Start of Text)	34	22	"	66	42	B	98	62	b
3	03	ETX	(End of Text)	35	23	#	67	43	C	99	63	c
4	04	EOT	(End of Transmission)	36	24	$	68	44	D	100	64	d
5	05	ENQ	(Enquiry)	37	25	%	69	45	E	101	65	e
6	06	ACK	(Acknowledgment)	38	26	&	70	46	F	102	66	f
7	07	BEL	(Bell)	39	27	'	71	47	G	103	67	g
8	08	BS	(Backspace)	40	28	(	72	48	H	104	68	h
9	09	HT	(Horizontal Tab)	41	29	)	73	49	I	105	69	i
10	0A	LF	(Line Feed)	42	2A	*	74	4A	J	106	6A	j
11	0B	VT	(Vertical Tab)	43	2B	+	75	4B	K	107	6B	k
12	0C	FF	(Form Feed)	44	2C	,	76	4C	L	108	6C	l
13	0D	CR	(Carriage Return)	45	2D	-	77	4D	M	109	6D	m
14	0E	SO	(Shift Out)	46	2E	.	78	4E	N	110	6E	n
15	0F	SI	(Shift In)	47	2F	/	79	4F	O	111	6F	o
16	10	DEL	(Data Link Escape)	48	30	0	80	50	P	112	70	p
17	11	DC1	(Device Control 1)	49	31	1	81	51	Q	113	71	q
18	12	DC2	(Device Control 2)	50	32	2	82	52	R	114	72	r
19	13	DC3	(Device Control 3)	51	33	3	83	53	S	115	73	s
20	14	DC4	(Device Control 4)	52	34	4	84	54	T	116	74	t
21	15	NAK	(Negative Acknowledgement)	53	35	5	85	55	U	117	75	u
22	16	SYN	(Synchronous Idle)	54	36	6	86	56	V	118	76	v
23	17	ETB	(End of Trans. Block)	55	37	7	87	57	W	119	77	w
24	18	CAN	(Cancel)	56	38	8	88	58	X	120	78	x
25	19	EM	(End of Medium)	57	39	9	89	59	Y	121	79	y
26	1A	SUB	(Substitute)	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	(Escape)	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	(File Separator)	60	3C	<	92	5C	\	124	7C	|
29	1D	GS	(Group Separator)	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	(Record Separator)	62	3E	>	94	5E	^	126	7E	~
31	1F	US	(Unit Separator)	63	3F	?	95	5F	_	127	7F	DEL

Apéndice B. Cálculo del CRC Utilizando Tablas

A seguir es presentada una función, utilizando lenguaje de programación "C", que implementa el cálculo del CRC para el protocolo Modbus-RTU. El cálculo utiliza dos tablas para suministrar valores precalculados de los desplazamientos necesarios para la realización del cálculo. El algoritmo fue obtenido y es explicado en los documentos referenciados en el ítem 5.

```
/* Table of CRC values for high-order byte */
static unsigned char auchCRCHi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40 };

/* Table of CRC values for low-order byte */
static char auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04,
0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09, 0x08, 0xC8,
0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC,
0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3, 0x11, 0xD1, 0xD0, 0x10,
0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,
0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A, 0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38,
0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C,
0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26, 0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0,
0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4,
0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F, 0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,
0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C,
0xB4, 0x74, 0x75, 0xB5, 0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0,
0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54,
0x9C, 0x5C, 0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98,
0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80, 0x40 };

/* The function returns the CRC as a unsigned short type */
unsigned short CRC16(puchMsg, usDataLen)
unsigned char *puchMsg;                  /* message to calculate CRC upon    */
unsigned short usDataLen;                /* quantity of bytes in message     */
{
    unsigned char uchCRCHi = 0xFF;       /* high byte of CRC initialized     */
    unsigned char uchCRCLo = 0xFF;       /* low byte of CRC initialized      */
    unsigned uIndex;                     /* will index into CRC lookup table */
    while (usDataLen--)                  /* pass through message buffer      */
    {
        uIndex = uchCRCLo ^ *puchMsgg++; /* calculate the CRC                */
        uchCRCLo = uchCRCHi ^ auchCRCHi[uIndex};
        uchCRCHi = auchCRCLo[uIndex];
    }

    return (uchCRCHi << 8 | uchCRCLo);
}
```

Apéndice C. Cálculo del CRC Utilizando Desplazamiento de Registradores

En este ítem es descrito el algoritmo para el cálculo del CRC utilizado en la comunicación Modbus-RTU, a través del desplazamiento de registradores. El algoritmo fue obtenido y es explicado en los documentos referenciados en el ítem 5.

El cálculo del CRC es iniciado primeramente cargándose una variable de 16 bits (referenciado a partir de ahora como variable CRC) con el valor FFFFh. Después se ejecuta los pasos de acuerdo con la siguiente rutina:

1. Someterse el primer byte del mensaje (solamente los bits de datos - start bit , paridad y stop bit no son utilizados) a una lógica XOR (O exclusivo) con los 8 bits menos significativos de la variable CRC, retornando el resultado en la propia variable CRC.
2. Entonces, la variable CRC es desplazada una posición a la derecha, en dirección al bit menos significativo, y la posición del bit más significativo es rellenada con 0 (cero).
3. Luego de este desplazamiento, el bit de *flag* (bit que fue desplazado para fuera de la variable CRC) es analizado, ocurriendo lo siguiente:
   - ☑ Si el valor del bit es 0 (cero), nada es hecho
   - ☑ Si el valor del bit es 1, el contenido de la variables CRC es sometido a una lógica XOR con un valor constante de A001h y el resultado es retornado a la variable CRC.
4. Se repite los pasos 2 y 3 hasta que los ochos desplazamientos tengan sido hechos.
5. Se repite los pasos de 1 a 4, utilizando el próximo byte del mensaje, hasta que todo el mensaje tenga sido procesado.

El contenido final de la variable CRC es el valor del campo CRC que es transmitido en el fin del telegrama. La parte menos significativa es transmitida primero (CRC-) y en seguida la parte más significativa (CRC+).